

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA KHOA HỌC MÁY TÍNH

MÁY HỌC

BÁO CÁO ĐỒ ÁN MÔN HỌC
CHỦ ĐỀ BÁO CÁO
HUẤN LUYỆN MODEL CHO
FACE MAKEUP APP

Giáo viên hướng dẫn: Lê Đình Duy

Phạm Nguyễn Trường An

Sinh viên thực hiện: 17521191 – Vũ Ngọc Trường

16521503 – Trần Khánh Hà

19521663 – Nguyễn Duy Khang

8/17/2021

Mục lục

Chương 1. Tổng quan

- 1.1. Mục tiêu
- 1.2. Mô tả bài toán
- 1.3. Mô tả dữ liệu

Chương 2. Các nghiên cứu trước

Chương 3. Xây dựng dữ liệu

Chương 4. Training và đánh giá model

- 4.1. Thuật toán
- 4.2. Training
- 4.3. Đánh giá
- 4.4. Kết luận

Chương 5. Ứng dụng và Hướng phát triển

- 5.1. Ứng dụng
- 5.2. Đánh giá ứng dụng
- 5.3. Hướng phát triển

Chương 1. Tổng quan:

1.1 Mục tiêu:

- Trong bản báo cáo này chúng tôi nhắm đến việc so sánh 2 model mà nhóm đã huấn luyện shape_predictor_68_face_landmark.dat và shape_predictor_20_lip_landmark.dat.
- Từ đó dựa trên performance của chúng mà quyết định sử dụng cho 2 ứng dụng nhóm đang xây dựng là face-makeup và lip-coloring.
- Đưa ra kết luận của nhóm về vấn đề horizontal or vertical (chiều sâu hay chiều rộng) trong việc xây dựng bộ dữ liệu và huấn luyện model

1.2 Mô tả bài toán :

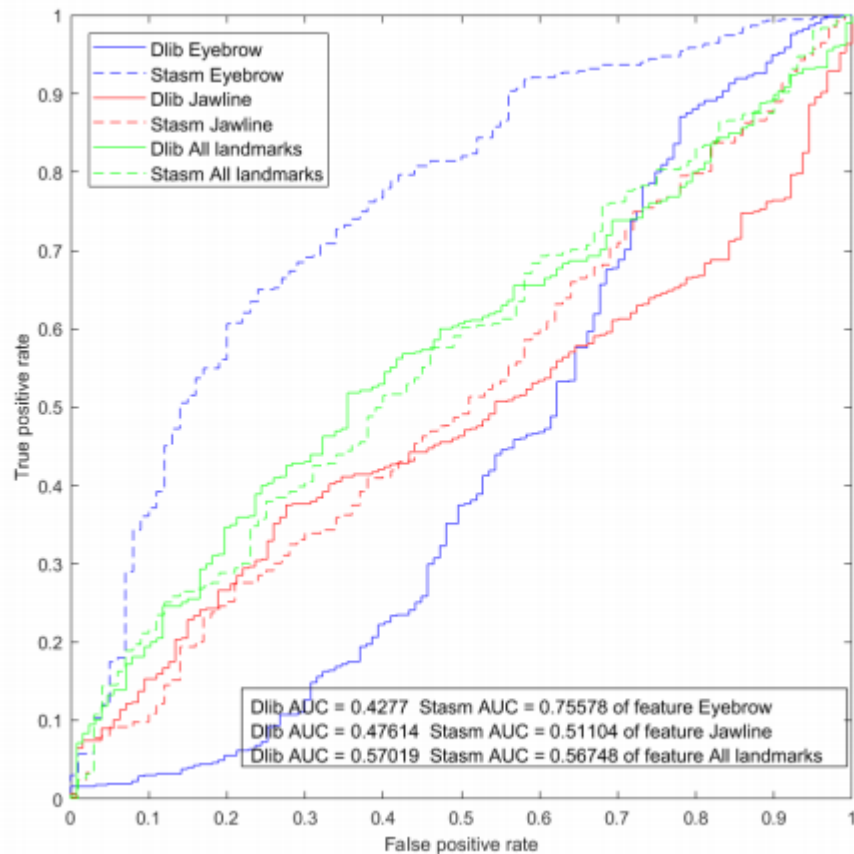
- Bài toán chọn hình ảnh là chân dung của người dùng lấy từ camera của các thiết bị ở đây nhóm chọn là camera laptop
- Model Input : là ảnh trong đó sẽ có chân dung của ít nhất 1 người
- Model Output : sẽ là ảnh với các landmarks của khuôn mặt được model dự đoán
- Đối với bài toán face-makeup input sẽ gồm ảnh chân dung của người sử dụng cho vào và ảnh chân dung đã được makeup mà người sử dụng muốn makeup giống như vậy , output sẽ là ảnh chân dung người sử dụng đã được makeup
- Đối với bài toán lip-coloring input sẽ là ảnh chân dung người sử dụng cho vào và output sẽ là ảnh chân dung với môi được tô màu theo ý người sử dụng
- Output của model sẽ được sử dụng kết hợp với các phương pháp của lĩnh vực computer vision để cho ra kết quả ứng dụng
- Từ yêu cầu ta xác định được bài toán gồm 2 bước :
 - Face detection : xác định khuôn mặt (sử dụng model của dlib)
 - Shape predictor : xác nhận vị trí landmarks của khuôn mặt (train)

1.3 Mô tả dữ liệu :

- Dữ liệu gồm chân dung của nhiều người ở nhiều góc độ biểu cảm khác nhau và thông tin đặc trưng của chân dung ở đây là 68 điểm landmarks
- Sự phát triển của mạng xã hội , trend dẫn đến lượng lớn data là ảnh , chân dung người điều khó khăn trong dữ liệu là xác định 68 điểm landmarks của ảnh
- Chúng tôi thấy rằng việc thu thập dataset mới là cần thiết vì các dataset hiện tại mang tính chiều rộng nên khi áp dụng vào thực tế chúng ta cần những bộ dataset mang tính chiều sâu để phù hợp với nhu cầu sử dụng
- Và khi thu thập thêm data ta có thể tăng độ chính xác cho model và làm giàu nguồn tài nguyên cho những người đi sau

Chương 2 . Các nghiên cứu trước

- Phương pháp sử dụng : DLIB và STASM
- Bộ dữ liệu : FRGCV2 và SCFACE
- Performance :






















	DLIB					Average Dlib
	Eye	nose	Eyebrow	Jawline	All land-marks	
HQ-HQ	0,72	0,78	0,78	0,85	0,91	0,81
HQ-MQ	0,61	0,71	0,64	0,71	0,79	0,69
Drop in %	51,6	26,6	49,4	39,7	29,9	38,1

	STASM					Average STASM
	Eye	nose	Eyebrow	Jawline	All land-marks	
HQ-HQ	0,69	0,73	0,78	0,80	0,88	0,77
HQ-MQ	0,57	0,61	0,59	0,63	0,70	0,62
Drop in %	64,0	50,9	67,0	55,2	48,3	56,2

- Kết quả đạt được chưa cao có khả năng là do bộ dữ liệu dùng để train vẫn còn nhỏ . Nếu tăng bộ dữ liệu lên thì có thể tăng performance của model

Chương 3 . Xây dựng dữ liệu

- Bộ dữ liệu được nhóm sử dụng là bộ dataset tổng hợp từ nguồn : http://dlib.net/files/data/ibug_300W_large_face_landmark_dataset.tar.gz
- Sau khi thử dùng để train model trên colab và thất bại nhóm đã giảm số lượng data xuống còn 1 nửa
- Bộ data trên được dùng để train shape_predictor_68_face_landmarks.dat
- Sau đó từ bộ data trên nhóm đã xây dựng lại bộ data mới để train model shape_predictor_20_lip_landmarks.dat bằng hàm parse.py để tạo ra 2 file label_lip_train.xml và label_lip_test.xml

Name ↓	Owner	Last modified	File size
 .ipynb_checkpoints	me	Aug 6, 2021 me	—
 train.xml 	me	Aug 8, 2021 me	9.4 MB
 shape_predictor.py 	me	Aug 8, 2021 me	807 bytes
 performance.py 	me	Aug 7, 2021 me	581 bytes
 parse.py 	me	4:22 PM me	720 bytes
 labels_lips_train.xml 	me	Aug 8, 2021 me	3.1 MB
 labels_lips_test.xml 	me	Aug 7, 2021 me	947 KB
 labels_ibug_300W.xml 	me	Jul 10, 2021 me	21.2 MB
 labels_ibug_300W_train.xml 	me	Jul 10, 2021 me	18.4 MB
 labels_ibug_300W_test.xml 	me	Jul 10, 2021 me	2.8 MB

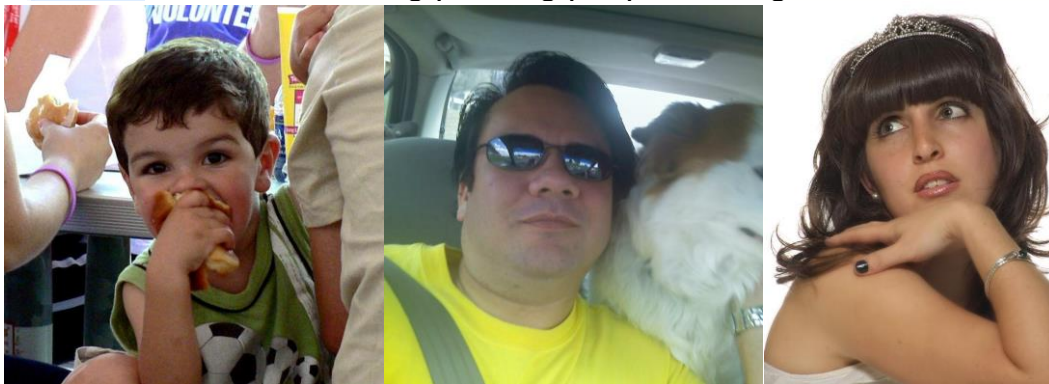
- Data train : 3192 - 76.15%
- Data test : 1000 - 23.85%

	Thiếu mắt	Thiếu mũi	Thiếu môi	Thiếu lông mày	Đầy đủ
Data_train	89	0	10	289	2812
Data_test	53	2	19	105	852

*Lưu ý : những trường hợp thiếu nhiều yếu tố thì nhóm tách riêng ra nên tổng sẽ không bằng 3192 và 1000



- **Dữ liệu khó xử lý : 29** . Là những bức ảnh mà các phần trên khuôn mặt bị che khuất hoàn toàn vô cùng khó để dùng đánh dấu landmark bằng phương pháp thủ công



Chương 4 . Training và đánh giá model

4.1 Thuật toán

Thuật toán `train_shape_predictor` của `dlib` thực hiện qua các bước sau :

- Lấy đặc trưng của ảnh của input
- Cho đặc trưng qua Ensemble of regression tree (ERT)
- Kết quả dự đoán của ERT sẽ được tăng độ chính xác thông qua nhiều lượt hồi quy (cascade of regressor)
- Ensemble : phương pháp học tăng cường . Mục tiêu là tạo ra mô hình dự đoán mạnh hơn dựa trên các mô hình yếu hơn ở đây là regression tree
- Regression Tree: là 1 dạng của cây quyết định thuộc loại thuật toán Classification and regression tree (CART) . Regression tree hay cây hồi quy ước lượng ra giá trị là số thực trong bài toán này là tọa độ x,y của các landmarks trên khuôn mặt . Các node của trong cây được xác định bằng phương pháp Gini (Gini impurity)

4.2 Training

- Nhóm xây dựng 2 model là `shape_predictor_68_face_landmark.dat` và `shape_predictor_20_lip_landmark.dat`.
- Sau khi tìm hiểu nhóm đã áp dụng thuật toán có trong thư viện `dlib` (`dlib.train_shape_predictor`)
- Thuật toán đòi hỏi 3 argument là file train , file model và các options .

+ Mã + Văn bản

```
import multiprocessing
import dlib
import os

faces=r'/content/drive/MyDrive/Do_an_MH/ibug_300W_large_face_landmark_dataset'
dat_file="/content/drive/MyDrive/Do_an_MH/shape_predictor_20_lip_landmarks.dat"
XML_set='/content/drive/MyDrive/Do_an_MH/ibug_300W_large_face_landmark_dataset/labels_lips_train.xml'

print("[INFO] setting shape predictor options...")
options = dlib.shape_predictor_training_options()
options.tree_depth = 4
options.nu = 0.1
options.cascade_depth = 15
options.feature_pool_size = 400
options.num_test_splits = 30
options.oversampling_amount = 5
options.oversampling_translation_jitter = 0.1
options.be_verbose = True
options.num_threads = multiprocessing.cpu_count()
print("[INFO] shape predictor options:")
print(options)
# train the shape predictor
print("[INFO] training shape predictor...")
training_xml_path = os.path.join(faces,XML_set)
dlib.train_shape_predictor(training_xml_path,dat_file, options)
```

	Train time	Test time	Size
Shape_predictor_68_face_landmark	12p4s	25p+7p	94.9MB
shape_predictor_20_lip_landmark	7p56s	1p+27s	29.1MB

- Qua quá trình tìm hiểu và thử nghiệm nhóm đã cài đặt 1 số options cho phù hợp với thực tế của nhóm trong đó có 6 options nhóm cho là quan trọng
- Options.tree_depth : chiều sâu của cây. Giá trị này ảnh hưởng đến độ chính xác và tốc độ của model chiều sâu của cây = $2^{\text{tree_depth}}$
- Options.nu : là regularization parameter . Giá trị này nằm trong khoảng $[0,1]$ càng gần 1 thì khả năng dự đoán trên tập train càng cao dễ dẫn đến overfitting càng gần 0 thì khả năng dự đoán trên thực tế càng cao nhưng cần nhiều data để huấn luyện
- Options.cascade_depth : là số lượng cascades để huấn luyện model . Ảnh đến độ chính xác và kích thước của model
- Options.feature_pool_size : là số lượng pixel dùng để trích xuất đặc trưng ở mỗi cây ngẫu nhiên (random trees) trong mỗi lượt (cascade) . Ảnh hưởng đến độ chính xác và tốc độ model
- Options.num_test_splits : Giá trị này chịu trách nhiệm tìm ra đặc trưng tốt nhất (feature) ở mỗi lượt (cascade) . Ảnh hưởng đến thời gian training và độ chính xác của model
- Options.oversampling_amount : là số lượng data augmentation áp dụng vào train data . Tăng số lượng data train tuy nhiên ảnh hưởng nhiều đến tốc độ train .

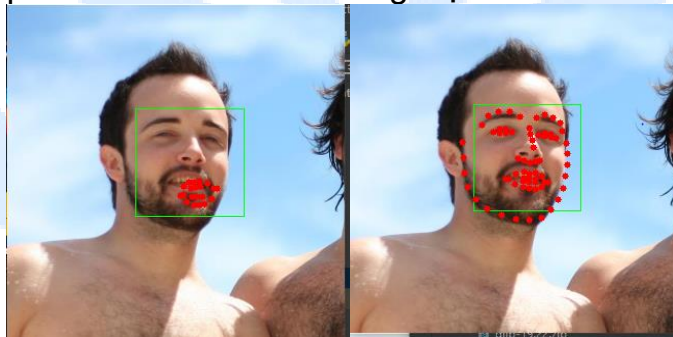
4.3 Đánh giá

	MAE (train set)	MAE (test set)
Shape_predictor_68_face_landmark	3.47	11
Shape_predictor_20_lip_landmark	3.25	12.7

- Model shape_predictor_68_face_landmark (model 1) và model shape_predictor_20_lip_landmark (model 2) không bị overfitting
- Độ chính xác của 2 model trên tập test còn chưa cao nguyên nhân là do bộ data train còn ít chỉ khoảng 3000 bức ảnh
- Model 2 có độ chính xác thấp hơn model 1 đi ngược lại với dự đoán của nhóm



- Với 3 bức đầu khi môi bị che 1 phần model đánh giá sai vị trí phần còn lại của môi do phần che khuất môi có hình dáng tương đồng dẫn đến việc model predict sai → ta có thể cho thêm data về các trường hợp này để model có thể đưa ra dự đoán chính xác hơn
- Với 2 bức cuối cùng chúng ta đã thất bại trong việc xác định khuôn mặt dẫn đến model không thể đưa ra dự đoán → pre-train model face detector của dlib hoặc sử dụng 1 model khác có performance tốt hơn trong việc detect khuôn mặt



- Trong trường hợp này model 2 đã đưa dự đoán sai so với model 1
- Vì đây là môi 1 trong những facial landmarks chúng có số quy luật như môi phải thẳng hàng với mũi và ở dưới mũi vì model 2 không học về những phần khác của khuôn mặt nên hiển nhiên không thể đưa ra dự đoán đúng trong trường hợp này.

4.4 Kết luận

- Model shape_predictor_68_face_landmarks.dat (model1) có tầm ứng dụng cao hơn model shape_predictor_20_lip_landmarks.dat (model2) khi có thể áp dụng cho cả 2 app
- Tuy nhiên model 2 lại vượt trội hơn về mặt dung lượng và tốc độ. Dataset model 2 sử dụng cũng tốn ít hơn vì khu vực cần dự đoán ít hơn nên có khả năng cao sẽ thu thập được bộ data đầy đủ hơn về các trường hợp khác nhau
- Qua đây nhóm củng cố được suy nghĩ của mình về vấn đề sử dụng và xây dựng bộ data chuyên sâu hơn bám sát vào thực trạng yêu cầu của từng bài toán sẽ giúp ta đạt được kết quả tốt hơn là sử dụng liên tục 1 bộ data có sẵn



Chương 5. Ứng dụng và Hướng phát triển

5.1. Ứng dụng:

5.1.1. Face_makeup:

- Phát hiện và khoanh vùng khuôn mặt : 68 landmarks + 11 landmark vùng trán được mở rộng thêm
- Facial alignment : sử dụng Delaunay Triangulation
- Layer Decomposition and Masking
- Color and Detail Transfer
- Reducing sharp edges : cắt tỉa góc cạnh để ảnh mượt hơn thực tế hơn

5.1.2. Lip_coloring:

- Crop chính xác phần môi : cv2.fillPoly
- Color : Điều chỉnh màu muốn tô
- Gaussian blur : khiến cho vùng được tô màu bớt góc cạnh và nhìn tự nhiên hơn

5.2. Đánh giá ứng dụng:

- Kết quả:
 - Tạo được app để dễ sử dụng hơn
 - Giao diện app đơn giản để sử dụng
- Hạn chế:
 - Giao diện app chưa đẹp
 - Còn nhiều hạn chế khi ảnh bị che khuất mặt bị biến dạng
 - Dataset thiếu độ đa dạng chưa thoả hết mọi kiểu trang điểm khuôn mặt
 - Thời gian thực thi ứng dụng không nhanh như dự kiến

5.3. Hướng phát triển:

- Có thể áp dụng nhiều trong thực tế ví dụ như là : nhận dạng khuôn mặt , nhận diện khuôn mặt , deepfake , làm filter, v. v. v.
- Nhóm áp dụng model train được để xây dựng thành công 2 ứng dụng face_makeup và lip_coloring
- Tiếp theo nhóm mong có thể hướng ứng dụng của mình đến người sử dụng trên nhiều nền tảng khác nhau và dựa trên hướng suy nghĩ phát triển ra các model chuyên sâu như mắt và mũi lông mày . Và có thể cân bằng 3 yếu tố kích thước , tốc độ và độ chính xác tốt hơn nữa

Tham khảo

https://www.csc.kth.se/~vahidk/face_ert.html

<https://ibug.doc.ic.ac.uk/resources/300-W/>

<https://medium.datadriveninvestor.com/training-alternative-dlib-shape-predictor-models-using-python-d1d8f8bd9f5c>

http://essay.utwente.nl/77533/1/Bachelorpaper_Wouter_Pool.pdf

http://dlib.net/files/data/ibug_300W_large_face_landmark_dataset.tar.gz

https://github.com/Luca96/dlib-minified-models/blob/master/face_landmarks/training_script.py#L32

http://dlib.net/train_shape_predictor.py.html

http://dlib.net/train_shape_predictor.py.html

